



# Porting Unix Apps to Mac OS X

## Session 105





# Porting Unix Apps to Mac OS X

**Jason Evans, Dave Zarzycki**  
**Software Engineers**

# Introduction

- Porting Unix apps to Mac OS X
- Unique or unusual aspects of Mac OS X from a Unix perspective that developers face when porting existing Unix software



# Topics Not Covered

- Drivers/IOKit
- GUI
- Multimedia
- Printing
- Font handling
- Anything you can see

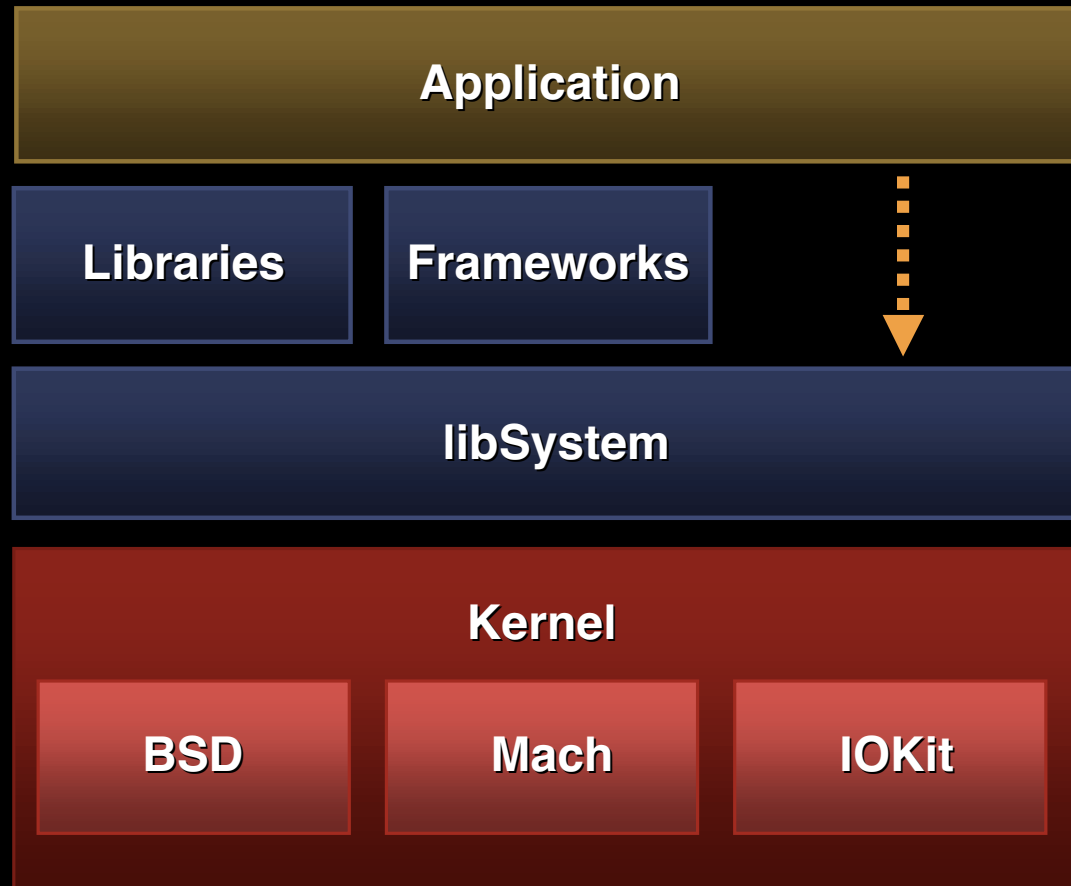


# History

- BSD
- Mach
- NeXTStep
- Mac OS X



# Basic System Diagram



# Packaging

- Distributions
  - Mac OS X
  - Mac OS X Server
  - Darwin
- Ports
  - Fink
  - Working on infrastructure like FreeBSD's



# Filesystem Portability

- POSIX
  - Case sensitive
  - Sparse files (not universal)
- HFS +
  - Case insensitive, case preserving
  - Resource forks
- Hardlinks not supported by all filesystems





# Libraries

- Mach-O binary format
  - Versioning
  - Plug-ins
  - Bundles vs. libraries
- Two-level namespace
- Static linking seldom used
- Mac OS 9-style weak symbols
- Frameworks
- libSystem



# Runtime Issues

- Configuration and resource files
  - OpenDirectory for system preferences
  - CF/NSPreferences for user preferences
- System startup
  - `/System/Library/StartupItems`
  - Bundle based, but not CFBundles
  - Composed of an executable and a plist



# More Runtime Issues

- Prebinding
  - Pro: decreases run-time startup cost
  - Con: modifies executables and libraries
    - Consequently confusing some applications such as backup and security tools
- `/bin/sh` is `bash`, was `zsh`
  - Portable shell scripts should not notice or care



# Frameworks

- Bundle-based alternative to the Unix hierarchy
- Contains libraries and headers
- Potentially contains other resources
- Can live in the following locations
  - `/System/Library/Frameworks`
  - `/Library/Frameworks`
  - `/Network/Library/Frameworks`
  - `$HOME/Library/Frameworks`
  - The application itself



# Authentication/Authorization

- Security framework
  - Apple's preferred API
  - Capability/rights-based system
- PAM available for compatibility purposes



# Development Environment

- Mostly GNU toolchain
  - gcc
  - gdb
  - make
- cpp precompiled headers
  - --no-cpp-precomp
- Project Builder
- Interface Builder



# API Considerations

- poll(2) missing
- pthreads partially implemented
- Asynchronous I/O missing
  - We do have pread()/pwrite()
- OpenSSL is available, CDSA is preferred
- Objective C
- libtool vs. GNU libtool (glibtool)



# “Mach”ing Your Code

- Process priorities
- Bootstrap ports
- `ptrace(2)` only implements “attach”
  - Use Mach APIs for other `ptrace()`-like functionality





# Mailing Lists

- Apple (<http://lists.apple.com>)
  - apple-cdsa
  - unix-porting
  - darwin-development
  - darwinos-users
  - darwin-kernel
  - darwin-userlevel



# Web sites

<http://www.opensource.apple.com/>

<http://www.opendarwin.org/>

<http://fink.sourceforge.net/>



# Books

- Mac OS X for Unix Developers (O'Reilly)  
<http://developer.apple.com/techpubs/>
  - UNIX Porting Guide
  - Inside Mac OS X: Mach-O Runtime Architecture
  - Additional developer toolchain documents



# Roadmap

---

## **107 The Darwin Kernel:**

The core of Mac OS X

Civic

**Wed., 9:00am**

---

## **110 Security:**

Authorization in Mac OS X

Civic

**Wed., 2:00pm**

---

## **FF002 Feedback Forum:**

Darwin

Room J1

**Wed., 3:30pm**



# Who to Contact

---

**Rob Braun**

[bbraun@apple.com](mailto:bbraun@apple.com)

---

**Jason Evans**

[jevans@apple.com](mailto:jevans@apple.com)

---

**Jordan Hubbard**

[jkh@apple.com](mailto:jkh@apple.com)

---

**Dave Zarycki**

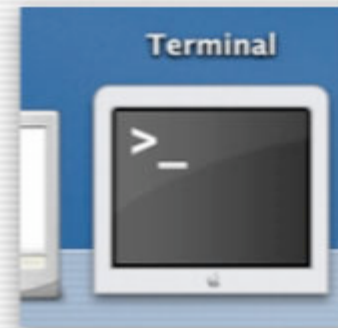
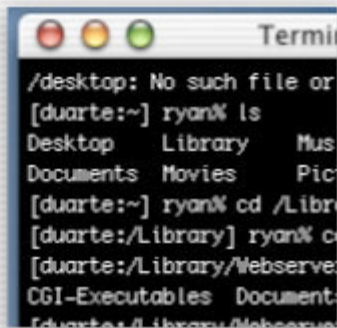
[zarycki@apple.com](mailto:zarycki@apple.com)

---





# Q&A



**Jason Evans, Dave Zarzycki**  
**Software Engineers**  
**jevans@apple.com, zarzycki@apple.com**

<http://developer.apple.com/wwdc2002/urls.html>

 **WWDC2002**

 **WWDC2002**



 **WWDC2002**